
 論 文

大韓造船學會論文集
 第 29 卷 第 4 號 1992 年 11 月
 Transactions of the Society of
 Naval Architects of Korea
 Vol. 29, No.4, November 1992

객체지향 개념과 가시화 기법에 의한 선박 개념설계용
 그래픽 사용자 인터페이스 모델

이경호*, 한순홍*, 이동곤*, 이규열*

Object-Oriented Graphical User Interface Model
 for Visualization of Ship Conceptual Design

by

Kyung Ho Lee*, Soon Hung Han*, Dong Kon Lee* and Kyu Yeul Lee*

요 약

컴퓨터 그래픽스와 빠른 하드웨어의 발달에 힘입어, WYSIWYG(What You See Is What You Get)로 특징지어지는 GUI(Graphical User Interface)는 기존의 사용자 인터페이스 개념을 컴랜드를 이용한 언어모델에서 풀다운 메뉴, 스크롤바, 아이콘 등을 이용한 공간모델로 바꾸어 놓았다. 여기서 개발된 선박개념설계를 위한 그래픽 사용자 인터페이스 모델(MBASWIN)은 Event-driven 프로그래밍 기법과 객체지향 개념을 도입하였다. 또한 기존의 설계 프로그램이 인터페이스 부분을 포함하여 프로그램 전체를 제어했던 방식에서 벗어나, 설계 프로그램과 사용자 인터페이스 부분을 완전 분리하여, 사용자 인터페이스 루틴이 설계 루틴들을 제어하는 방식을 도입하였다. 이를 이용하여 설계자는 유연성있는 설계를 수행할 수 있게 된다.

Abstract

By the help of computer graphics and rapid development of hardwares, GUI(Graphic User Interface) represented by WYSIWYG(What You See Is What You Get) changed user interface from linguistic model(e.g : command, etc.) to spatial model(e.g : pulldown menu, scroll bar, icon, etc.). This graphical user interface model for the ship conceptual design(MBASWIN) adopted event-driven programming technique and object-oriented concepts. Different from traditional design programs, MBASWIN is separate from the design program completely and controls all design modules. This enable a designer to accomplish a flexible designs.

발 표 : 1991년도 대한조선학회 추계연구발표회('92. 11. 16.)
 접수일자: 1992년 3월 16일, 재접수일자: 1992년 7월 7일
 * 정회원, 해사기술연구소

大韓造船學會論文集 第29卷 4號 1992年 11月

1. 서 론

1980년대 부터 널리 보급되기 시작한 Engineering Workstation(이하 EWS라 함)은 대체로 UNIX 운영 체제(O/S)를 바탕으로 하고 있고, EWS의 장점인 그래픽 기능을 이용한 윈도우시스템이 개발되어 점차 과거의 대형컴퓨터를 대신 하고 있다[1].

한편, 지난 10년간 소프트웨어는 非대화식 수행방법에서 대화식 수행방법으로 발전되어 왔다. 이와 같은 대화식 프로그램이 실행되는 과정에 대한 사용자의 직접적인 관여는, 어떻게 프로그램이 외부세계와 인터페이스 되는가에 대한 우리의 시각을 완전히 바꾸어 놓았다. 사용자 대화방식은 소프트웨어 엔지니어로 하여금 여러가지 인간적 요소(사용의 용이성, 정보의 표현력등)를 생각케 만든다. 이러한 관심은 그래픽 디스플레이, 메뉴를 이용한 입력, 마우스를 통한 선택 등을 위시한 여러가지 대화식 유형을 이끌어 냈다[2].

대화식 소프트웨어는 사용자 인터페이스를 독립적으로 사용하기 위해, 어플리케이션으로 부터의 독립을 강조하고 있는데(Fig.1 참조), 이것은 몇가지 유리한 측면을 가지고 있다.

첫째, 사용자 인터페이스를 함께 묶을 수 있는 요소들로 세분할 수 있다. 사용자는 이와같은 접근 방식의 기반이 되는 기능들의 자세한 사양을 이해하지 않고도, 이러한 요소들을 사용할 수 있다. 둘째, 사용자는 사용자 인터페이스를 요구조건이 유사한 다른 어플리케이션에서 다시 사용될 수 있도록 신속하게 수정할 수 있다. 셋째, 사용자는 어플리케이션

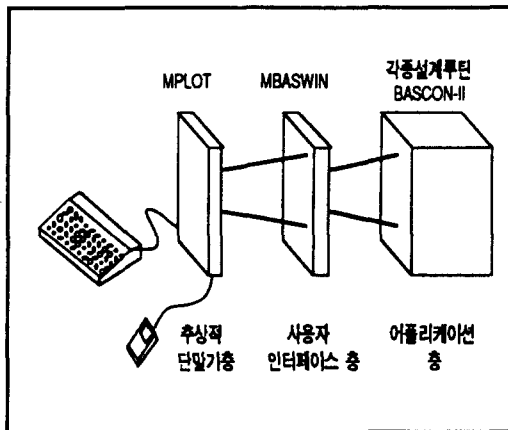


Fig.1 Independence of user interface from application

부분에 대해 어떤 영향도 끼치지 않고 인터페이스를 개발할 수도 있다.

이러한 장점들을 이용해 많은 원리들과 기법들이 나타났다. 여기서 원리들이란 재사용성과 캡슐화이다. 또, 기법이라 함은 신속한 프로토타이핑과 반복적인 개발이다. 대화식 그래픽 인터페이스 환경을 지원하기 위해 개발자에게 이러한 원리들과 기법을 지원하는 설계방법론과 개발환경이 필요하다. 이것이 바로 객체지향적 설계와 프로그래밍이 활용되는 환경이다.

선박의 개념설계 과정은 설계자의 개념을 전개하는 과정이고, 선박의 윤곽을 결정하는 중요한 과정이기 때문에 설계자의 창조력이 요구되며, 설계 방법은 상황변화에 원활히 대응할 수 있는 유연성을 가져야 한다. 또한 여러 종류의 복잡한 데이터를 처리하는 작업이 자주 발생하기 때문에 설계과정을 용이하게 지원하기 위하여 설계작업 자체를 가시화하는 것도 중요한 일이다. 또한 설계자 및 사용자들이 그 시스템에 대한 특별한 지식이 없이도 쉽게 접근할 수 있어야 하며, 프로그램의 높은 이식성을 위해서는 표준화된 방법을 사용하여 그래픽 사용자 인터페이스를 개발할 필요가 있다.

본 연구에서는 표준 운영체제(O/S)로 자리잡고 있는 UNIX 환경하의 X-Window 시스템에 기초한 Motif를 이용하여, 각각의 데이터 윈도우(Object)의 계층(Hierarchy)을 구축하였다. 또한 객체지향 개념의 잇점을 충분히 살리며 선박개념설계의 유연성을 증가시켜 주도록 전체적인 설계작업의 흐름을 풀다운 메뉴(Pulldown Menu)로 작성하였다. 그리고 기존에 FORTRAN으로 개발된 선박설계용 루틴과의 인터페이스를 통해 표준화된 Tool을 사용한 선박개념설계용 그래픽 사용자 인터페이스 모델을 개발하였다.

본 논문은 과학기술처의 지원을 받아 수행한 국가 특정 연구사업인 "선박설계·생산 전산 시스템(CSDP)-초기설계 시스템 개발" 연구사업의 3차년도 연구결과와 일부분이다.

2. X-Window 시스템과 객체지향 프로그램(OOP)

2.1 네트워크 지향 프로그램

근래에 EWS이 각광을 받고 있는 큰 이유중의 하나는 그래픽 기능이 뛰어나다는 것인데, 그 중에서도 윈도우 시스템이 큰 역할을 한다. 이 윈도우 시스템

템 중에서 미국의 MIT를 중심으로 개발된 X-Window 시스템이 현재 널리 쓰이고 있다[3].

X-Window는 네트워크를 근간으로 한 Window 시스템이라 할 수 있다. 즉 응용 프로그램과 화면의 디스플레이가 각각 다른 컴퓨터 시스템에서 수행될 수 있다는 것이다[4][5]. 이는 어떤 컴퓨터에서 운용되고 있는 응용 프로그램이 다른 컴퓨터에 있는 디스플레이 장치를 이용할 수 있으며, 이때 이 두개의 하드웨어는 같은 구조나 운영체제(O/S)를 가질 필요가 없다. 다시 말하면, 특정 기계에서만 동작하는 응용 프로그램도 사용할 수 있고, 사용자 인터페이스 부분을 여러 Language로 Coding할 수고를 덜어 줄 수 있다. 본 연구에서는 선박 개념설계 프로그램(FORTRAN으로 개발된 기존의 프로그램)과 그래픽 사용자 인터페이스 프로그램(C 언어로 개발 : MBASWIN)을 SUN W/S에서 개발하였다. 이는 SUN에서 개발되어 있는 프로그램을 따로 복사하거나 수정하지 않고 수행하여 네트워크 상에 연결되어 있는 Apollo EWS상에 결과를 디스플레이 할 수 있다. 이러한 기능이 가능토록 X-Window에서는 Client-Server 모델을 도입하였다.

1) 클라이언트(Client)와 서버(Server) 모델

X-Window 시스템의 특징은 앞서도 언급한 바와 같이 네트워크 지향이며, 어플리케이션 프로그램은 Device로 부터 독립적이어야 한다. 이를 실현하기 위한 핵심적인 개념이 Client-Server 모델이다. X-Window 시스템에서의 각종처리는 Client와 Server 라는 2개의 프로그램간에 분담되어 행해진다. Server 는 Window를 생성하거나, 스크린 상에 문자를 표시하거나 도형을 출력한다. 키보드나 마우스의 입력을 처리하는 것도 Server의 역할이다. 이처럼 Server 는 입·출력 장치의 제어를 직접 담당하고 있다. 한편 Client는 Window의 생성을 요구하거나, 도형의 출력등을 Server에 요구하며, 이 요구는 X Protocol이라는 Byte-stream에 의해 Server에 전송된다[3][6]. X Protocol(Request)을 받는 Server는 Client의 요구에 응해 화면상에 Window를 표시하거나 도형을 출력한다. 또 키보드 입력이나 마우스 버튼의 동작은 Server에 의해 탐지되어 이 정보가 X Protocol (Event)로써 Client에 전송된다. Client와 Server는 네트워크 상의 각기 다른 시스템에 분산시킬 수 있다. 물론 한 시스템에 공존도 가능하다. 이런 Client-Server 모델을 도식화하면 Fig.2와 같다.

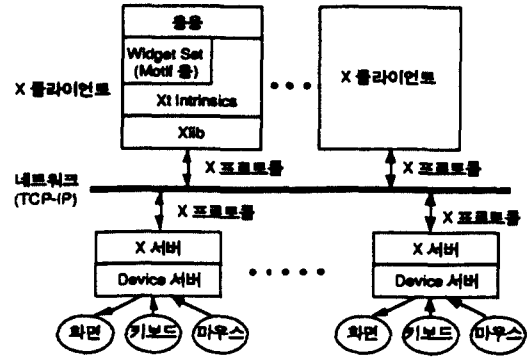


Fig.2 Client-server model

2.2 Event-driven 프로그래밍

그래픽 사용자 인터페이스의 프로그래밍은 기존의 절차식 프로그래밍 방법과 근본적으로 많은 차이가 있다. 기존의 절차식 프로그래밍에 기초한 문자 중심의 인터페이스(Character-Based Interface)는 프로그램이 한번 시작되면 모든 제어가 그 절차에 따른 제한을 받게 되어 유연성을 잃게 된다. 반면에 Event-driven 방식에 기초한 GUI는 사용자의 조작(Event)에 따라 유연하게 프로그램을 수행해 나갈 수 있다.

본 연구에서는 기존의 절차식 프로그래밍 방법에 의한 선박 설계 프로그램과 Event-driven 방식의 GUI(Graphic User Interface) 프로그램을 독립적으로 개발하여 연결하였는데, 기존의 선박설계 프로그램들이 Event-driven 방식의 GUI를 고려하지 않고 짜여져 있어서 연결하는데 많은 문제점을 갖고 있었다. 이에 대해서는 5장에서 자세히 설명하기로 한다.

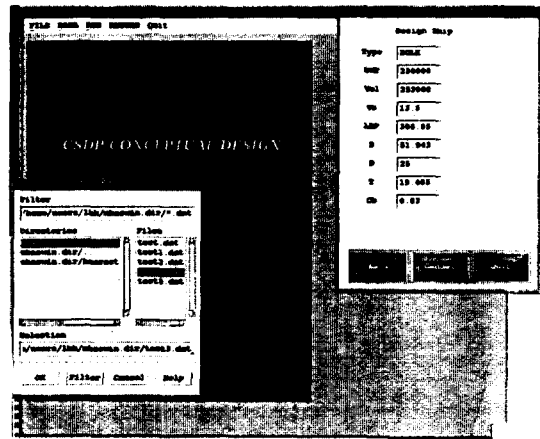


Fig.3 Incorporation both mouse and keyboard input

3. 표준화된 GUI개발을 위한 Motif

EWS 환경하에서 사용자 인터페이스를 개선하기 위해서 그래픽 사용자 인터페이스(GUI)를 도입하고 있으며, 그것은 Apple사의 맥킨토시 컴퓨터나 Microsoft 사의 Window 환경이 좋은 예이다. 컴퓨터 그래픽과 빠른 H/W의 개발에 힘입어 WYSIWYG (What You See Is What You Get)으로 대표되는 GUI는 사용자의 부담을 줄여주게 되었다. 즉 컴퓨터의 전통적인 사용자 인터페이스였던 명령어(Command)와 같은 언어모델(Linguistic model)로 부터 Pull-down 메뉴나 아이콘 등을 이용하는 공간모델(Spatial model)로 변환하게 된 것이다. 선박설계와 같이 문자적인 입력을 많이 필요로 하는 분야의 사용자 인터페이스 개발을 위해서는 이들 간의 적절한 조화가 필요할 것이다(Fig.3 참조)[7]. 버튼, 윈도우, 스크롤 바, Pull-down 메뉴, Icon 등 자주 쓰이는 그래픽 부품들의 클래스를 이용하여 프로그래머는

쉽게 GUI를 개발할 수 있다. 이러한 클래스들을 Widget Set이라고 하며, 이것은 X 라이브러리를 기반으로 하여 형성한 그래픽 객체(Object)로 이해할 수 있다. Fig.4는 선박개념설계용 그래픽 사용자 인터페이스로 개발된 MBASWIN에서, 설계자가 정한 DWT에 가장 가까운 실적선을 실적선 DB로부터 찾은 후 그 실적선의 선체 중앙부의 단면을 가시화한 예이다. 여기서 메뉴바, 버튼, 스크롤 바 등의 도형정보를 볼 수 있으며, 이들을 통해 사용자는 간편하게 설계작업을 수행해 나갈 수 있다. 이러한 사용자 인터페이스를 개발하는데 사용할 수 있는 Toolkit이나 Widget Set이 여러가지가 있으나, Motif를 최종적으로 결정하는 데는 그래픽 표준의 역할을 고려하였다. 현재 CSDP사업을 통하여 개발되고 있는 선박 개념설계 시스템은 완성되는 대로 국내의 여러 조선소에 이전될 것이다. 그러나 각 조선소의 전산시스템은 그 하드웨어나 소프트웨어환경이 많은 차이를 보인다. 이러한 이질적인 환경에 쉽게 이식될

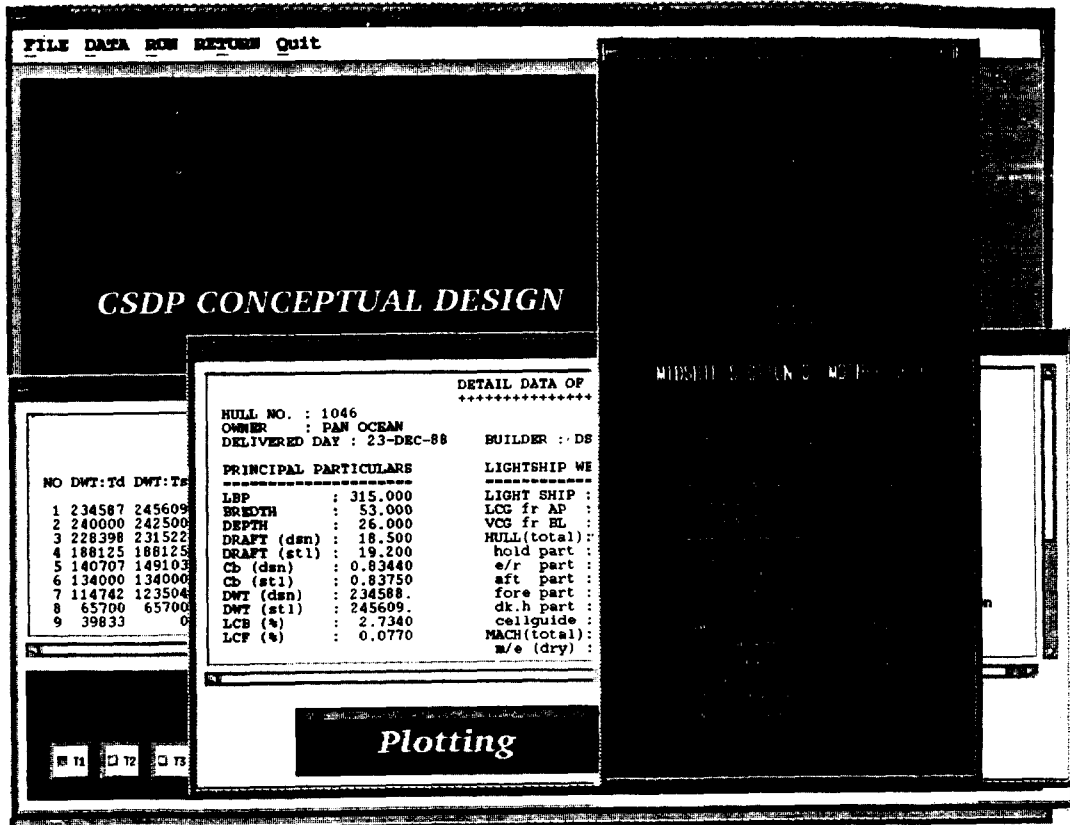


Fig.4 GUI for ship conceptual design by using graphical informations

수 있는 소프트웨어를 개발하기 위해서는 표준화된 시스템 소프트웨어를 사용하여야 한다. 그래픽 표준에는 공인된 표준(ISO, ANSI등), 산업계 표준(Industrial Standard), 실질적인 표준(De Facto Standard) 등이 있는데, Motif는 산업계 표준이라고 할 수 있다 [8].

따라서 하드웨어로 부터 독립적인 GUI를 개발하기 위해서는 UNIX에 바탕을 두고, 실질적인 표준 윈도우 시스템으로 자리잡고 있는 X-Window와 이를 바탕으로 한 Motif의 사용이 가장 좋은 것으로 생각되어 이를 중심으로 선박설계용 사용자 인터페이스를 개발하였다.

3.1 객체지향 개념에 의한 GUI 구현

X 라이브러리나 X Toolkit, Motif는 C언어와 UNIX의 범주안에서 절차식 언어이면서 객체지향 개념을 도입하고 있다. 객체지향 프로그래밍(Object-Oriented Programming: OOP)에서는 객체가 두 요소를 갖고 있다. 객체의 상태를 나타내는 데이터와 그 데이터를 읽어들이거나 쓰는 Method라 부르는 코드가 그것이다. '객체(Object)'를 이용한다는 것은 실제 세계 또는 Abstract Entity를 흉내내는 코드를 엮어내는 한 방법이다. 즉 프로그램 내에서 필요한 Widget(객체)들을 생성시키며, 이때 Widget들은 자기 고유의 Data Structure를 부여받게 되고 자기의 상위 객체로부터 필요한 Data를 상속받게 된다. 이렇게 생성된 Widget은 고유의 Method를 가질 수 있으며 어떤 Event에 대해서 Method에 해당하는 Function을 호출하여 프로그램을 수행하게 된다. 이러한 객체지향 개념을 도입한 그래픽 사용자 인터페이스는 유연한 설계 시스템을 구축하기 위해서는 필수적이라 할 수 있다.

3.2 Motif에서의 객체(Object)와 Method

OOP에서 객체가 두 요소를 갖고 있다는 한 예를 다음과 같은 실제 C프로그램의 일부를 이용하여 설명한다.

```
main(argc, argv)
int argc;
char *argv[ ];
{
.
.
}
```

```
button=XmCreatePushButton(toplevel, "button",
myArgs, ac);
XtAddCallback(button, XmNactivateCallback,
click_button, pass);
.
.
.
}
void click_button(...)
{
.
.
}
```

위의 간단한 예에서 볼 수 있듯이 button이라는 Widget(객체)을 생성할 때 myArgs라는 Data Structure에 여러가지 필요한 데이터를 주게 된다. 실제 프로그램에서의 데이터는 C언어의 Structure로 표현되며, Method는 Function에로의 Pointer로서 특정 Class를 위한 C언어에서의 Function들의 집합이다. 위의 예에서는 click_button이라는 Method가 button이라는 Widget(객체)에 등록되어서 그 버튼이 Click되었을 때 click_button이라는 함수가 작동하게 되어 있다.

3.3 Motif에서의 메시지(Message)

순수한 OOP에서는 객체끼리 메시지의 교환을 통해 통신을 하지만 Motif에서는 Function Calls, Events, Actions 및 Callback을 통해서 통신한다. XtSetValues와 XtGetValues라는 Function을 호출하여 Widget(객체)내의 자원(Resource) 값을 읽거나 쓸 수 있다. 또한 Widget은 응용 프로그램 뿐 아니라 사용자로부터의 각종 Event(마우스 클릭 등)에 대해 응답할 수 있다.

Widget은 어떤 Event가 발생했을 때 미리 등록되어진 Callback이라는 Action을 사용하여 지정해둔 루틴을 수행한다.

3.4 Motif에서의 캡슐화(Encapsulation)

Motif에서는 Widget 내부의 코드가 어떻게 되어있는지 응용 프로그래머는 알 필요없이 프로그램을 작성할 수 있다. 각각의 Widget Class는 프로그래머가 사용할 수 있는 Public 데이터와 그렇지 못한

Private 데이터를 갖고 있고 사용자는 프로그램 내에서 Public 데이터 타입을 서술한 헤더파일(header file)만 include 시키면 된다. 예를 들면 프로그램 시작부에 다음과 같이 선언하면 된다.

```
#include(Xm/PushB.h)
```

4. 선박 개념설계를 위한 그래픽 사용자 인터페이스(GUI) 개발

4.1 풀다운(Pulldown) 메뉴 방식을 이용한 선박 설계 Flow 제어

절차식 프로그래밍 기법에 의한 기존의 선박설계 프로그램들은 어플리케이션 루틴과 사용자 인터페이스 루틴이 혼합되어 있으며, 프로그램이 한번 시작되면 모든 제어가 그 절차의 제한을 받게 되어 유연성을 잃게 된다. 또한 데이터 입력 윈도우층(Layer)의 깊이가 깊어져서, 사용자가 프로그램 수행 도중 앞에 수행한 곳으로 다시 돌아가고자 할 때 여기에 유연하게 대처하기가 어렵다. 특히 선박설계와 같이 프로그램을 수행한 설계자가 그 중간 결과를 판단하여 앞서 수행했던 다른 루틴을 다시 수행하고자 할 때 프로그램이 이에 유연하게 대처해 갈 수 있어야 한다.

이러한 선박설계 고유의 특성을 고려할 때 데이터 입력 윈도우 층(Layer)의 깊이를 될 수 있는대로 줄여주며, 기존의 설계 프로그램이 인터페이스 부분을 포함하여 프로그램 전체를 제어했던 방식에서 벗어나야 한다. 이를 위해 설계 프로그램(어플리케이션 루틴)과 사용자 인터페이스 루틴을 완전 분리하여 사용자 인터페이스 루틴이 설계 루틴들을 제어하는 방식을 도입하여야 한다. 따라서 MBASWIN에서는 풀다운 메뉴 방식을 도입하여 각각의 메뉴를 그룹핑하였다. 그룹핑한 내용을 보면 Table 1과 같다.

이렇게 메뉴를 그룹핑 한 후, 각각의 메뉴에 대해서 대응되는 설계프로그램을 모듈 별로 분리하고 이것을 하나의 객체(object)로 하여, 사용자 인터페이스 루틴에서 이를 각 메뉴의 Call-back의 대상으로 구성하였다. 이렇게 함으로써 기존의 선박설계 프로그램이 갖고 있는 많은 불편한 점들을 개선하게 되었다.

MBASWIN의 풀다운 메뉴에서 사용된 객체로는 메뉴 바(Menu bar), 메뉴 버튼을 갖고 있는 메뉴 Pane, 메뉴 버튼 등이 있으며, 각 설계루틴이 Method 의 역할을 하고 있다.

Table 1 Menu grouping in MBASWIN

메뉴	sub 메뉴	용 도
FILE	New	새로운 데이터 파일 생성
	Load	기존의 데이터 파일을 로드시킴
Data	DB sorting	about DWT : 실적선 DB로 부터 사용자가 입력한 선박의 DWT와 가까운 순서대로 실적선 데이터를 Sorting 하여 가시화시켜줌
		about Vol
		:
	Detail View	: Sorting된 실적선 중에서 사용자가 선택한 배에 대한 자세한 정보를 그래픽과 함께 가시화시켜줌
	Data Set	: 설계자가 설계에 사용할 실적선의 자료를 Mohter Ship의 데이터로 저장함
Run	All routines	
	Engine selection	
	Compartment Division	

4.2 선박 개념설계 과정의 가시화

선박의 초기 윤곽을 결정하는 개념설계 과정은 상호영향을 주는 여러가지 인자들을 동시에 고려해야 하는 어려움과 함께, 그 결정을 뒷받침할 근거자료가 아직 마련되지 않다는 어려움이 같이 존재한다. 즉, 설계자는 결정을 내리기 위해서 여러가지 설계안에 대한 평가 결과가 필요하며, 그 평가결과를 얻기 위해서는 시험이 될 대상이 먼저 마련되어야 한다.

이러한 이유로 개념설계의 과정은 전통적으로 숙련된 기술자의 경험과 Know-How에 의존하여 왔다. 즉, 실적선 자료의 정리 정돈과 그 자료의 적절한 이용이 중요한 요소였다. 근래에 컴퓨터 분야의 빠른 발전에 힘입어 이 설계 과정을 전산화하려는 시도가 계속되고 있으나, 이 과정은 근본적으로 자동화하기 어려운 분야이며, 최대한 설계자의 판단 과정을 도와주는 도구들이 컴퓨터를 이용하여 개발되고 있다.

컴퓨터를 이용하여 설계자를 지원하는 데는 두가지 다른 전략이 있다. 그 한가지는 컴퓨터를 지능화시켜서 사람의 기능을 컴퓨터로 대체해 나가는 것으로 이것을 자동화의 목표라고 할 수 있다. 최적화 기법의 응용이나 인공지능 등이 이 종류에 속하는 것이다. 또 다른 전략은 인간이 하는 작업중에 컴퓨터가 더 잘할 수 있는 부분들은 컴퓨터 작업으로

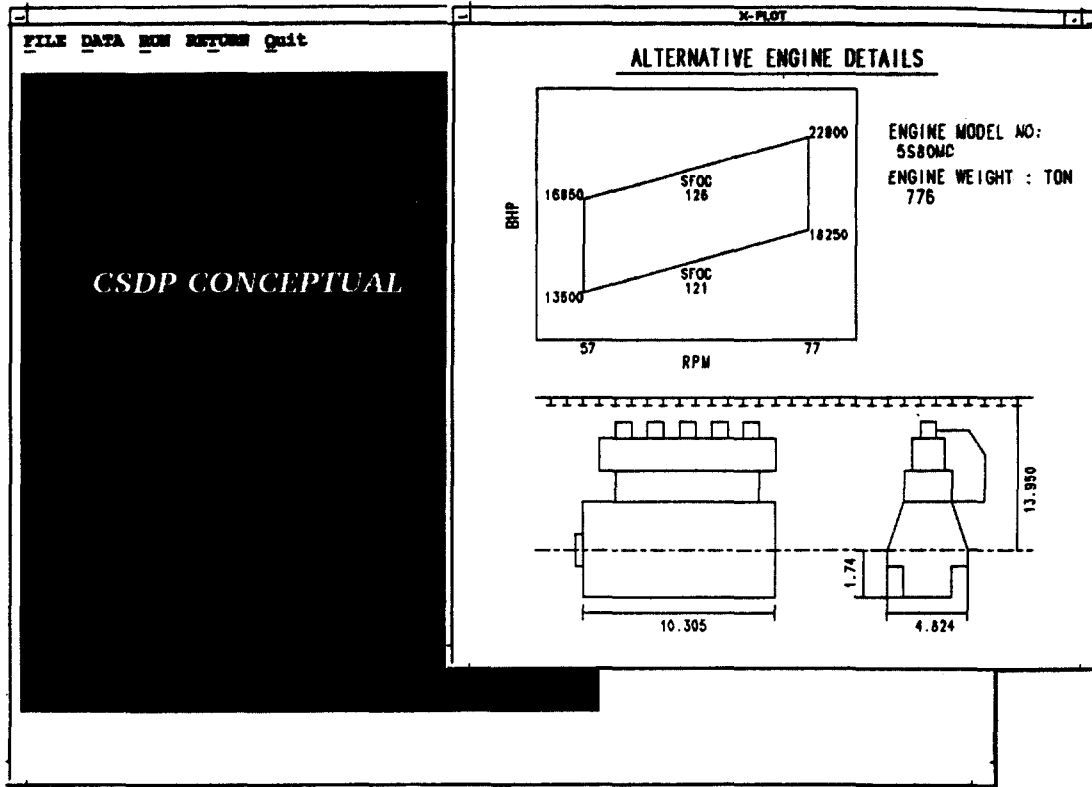


Fig.5 Visualization of optimum main engine selection

대체해 나가는 것으로, 인간의 작업환경을 가능한한 재현해 내는데 중점을 둔다. 이 종류에 속하는 것이 사용자 인터페이스 개선이나 가시화 분야이다[9].

이 글에서는 설계작업의 가시화를 통해서 사용자 인터페이스를 개선하는데 그 목표를 두었다. 설계작업의 가시화는 크게 두가지로 나눌 수 있는데, 첫째는 설계대상의 가시화이고, 둘째는 설계과정 자체의 가시화이다. 설계대상인 선박이라는 물체를 가시화하기 위하여 도면이라는 수단이 오랫동안 사용되어 왔으며, 이 도면작업을 전산화하는 것이 많은 CAD/CAM 시스템들의 역할이었다. 이 과정에서 컴퓨터 그래픽의 역할이 크며, 도면작업의 전산화와 설계대상의 가시화를 위해서는 3차원 그래픽이 중요한 역할을 한다. 이와 함께 사용자 인터페이스는 설계지원에 중요한 위치를 차지하고 있다. 여기서는 설계대상과 설계과정의 가시화를 위해서 X-Window에서 제공하는 2차원 그래픽 기능을 이용하였다. Fig.5에서 보인것은 설계과정에서 최적주기관 선정에 따른 설계 대상을 가시화한 예이다. 이런 작업들을

통해서 설계자는 손쉽게 작업 대상을 눈으로 확인하며 설계를 수행해 갈 수 있게 된다.

또한 데이터 입력의 가시화를 통해 사용자가 쉽게 설계에 필요한 데이터를 입력하도록 지원하였고 설계의 결과를 스크롤 바를 이용하여 화면의 크기에 상관없이 쉽게 볼 수 있도록 하였다. 이를 보여주는 것이 Fig.6, Fig.7이다.

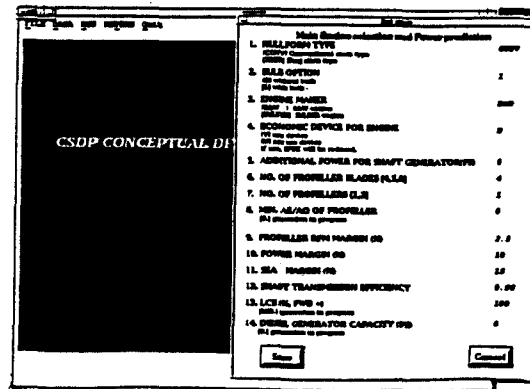


Fig.6 Visualization of data input

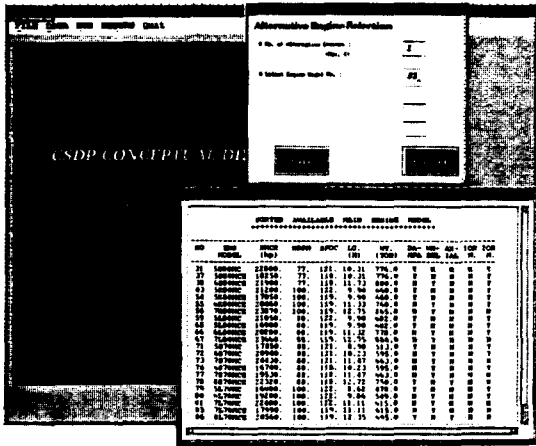


Fig.7 Visualization of output through scroll bar

5. 프로세스간의 통신(Inter-Process Communication)을 통한 데이터 흐름제어

2.2절에서도 언급한 바와 같이, 기존의 절차식 프로그래밍 방식을 따른 설계 프로그램과, Event-driven 방식을 갖는 Motif를 이용하여 만들어진 GUI 프로그램을 연결시키는 것은 쉬운일이 아니다.

MBASWIN에서는 Run버튼에 대한 한개의 Method로서 개념설계 전체의 흐름을 제어하는 모듈이 묶여있다. 이 모듈에서 각종 설계 서브모듈을 호출하게 된다. 그런데 이런 각 서브모듈 내에서 여러개의 데이터 입력을 위한 윈도우를 차례로 띄워서 데이터를 받아들여야 한다. 그러나 절차식 프로그래밍의 개념에서는 프로그래머가 원하는대로, 데이터를 받아들이는 동안 프로세스가 멈추어 있다가 입력이 종료되고 동시에 그 데이터를 바탕으로 계산을 계속해 나가는 것이 아니라, 윈도우가 떠서 데이터를 입력하는 동안 프로세스가 멈추지 않고 그 버튼에 묶여있는 Method루틴을 끝까지 수행해 버린다.

이러한 근본적인 프로그래밍 기법 차이에 의한 문제를 해결하기 위하여 MBASWIN에서는 UNIX에서 제공되는 프로세스간의 통신을 이용하여 설계 흐름을 제어하는 방식을 도입하였다.

UNIX에서는 실행 가능한 프로그램을 프로세스(process)라고 한다. UNIX는 사용자의 프로세스를 서비스하기 위한 여러가지 기능을 마련하고 있는데 이 기능을 시스템 호출이라고 한다[10]. 사용자는

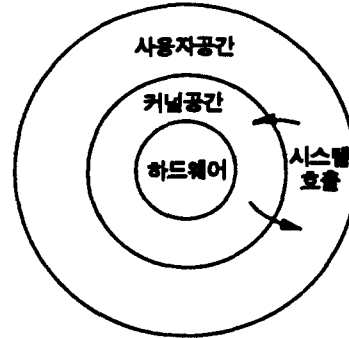


Fig.8 Concept of UNIX system call

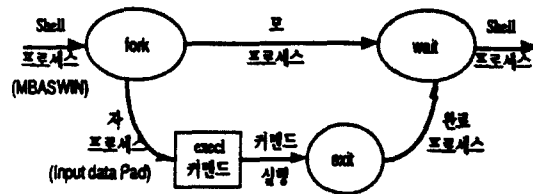


Fig.9 Inter-process communication concept in MBASWIN

프로그램 작성시에 UNIX의 어떤 기능을 서비스받기 위해서 단순히 서브루틴처럼 시스템 호출을 할 수 있다. Fig.8에서 처음 시스템 호출을 하면 제어가 사용자 공간에서 커널 모드로 옮겨지고 시스템 호출을 모두 마치면 제어는 다시 사용자 프로그램으로 복귀한다[11].

MBASWIN에서는 각 버튼에 등록된 Method 루틴에서 여러개의 데이터 입력을 위한 윈도우를 띄우고자 할 때, 현재의 프로세스를 멈추고 UNIX 시스템 함수인 fork(), execl(), wait() 등을 통해 또 다른 프로세스를 생성시킨 다음, 거기에서 윈도우를 만드는 루틴을 수행시킨다. 그 다음 데이터 입력을 하고 사용자가 특별한 Event를 줄 때 그 데이터들을 프로세스간의 통신 수단을 통해 기다리고 있던 모프로세스(Parent Process)에게 돌려주면서 모 프로세스가 계속 수행하도록 되어있다. 이를 간단히 도식화하면 Fig.9과 같다.

Fig.9의 내용을 MBASWIN에서의 실제 프로그램 예를 통해 설명하면 다음과 같다. 먼저 MBASWIN을 작동시키면 Fig.3, Fig.4와 같은 가시화된 윈도우를 통해 필요한 데이터를 마련할 수 있게 된다. 다음으로 메뉴바의 Run을 실행시키게 되면 GUI루틴(MBASWIN)이 전체적인 설계흐름을 제어하는

FORTTRAN 으로 작성된 서브루틴을 호출하게 되며, 여기서 각각의 계산루틴(Main Engine Selection, Compartment Division, Volume 계산 등)을 호출하여 설계를 수행해 나간다.

각각의 계산루틴들을 수행하기 위해서는 필요한 데이터를 마련해야 하는데, 설계자가 데이터를 쉽게 이해하고 수정할 수 있도록 Fig.6, Fig.7과 같은 데이터 입력 윈도우를 제공한다. 이때 데이터 입력 윈도우를 가시화하기 위해서 UNIX 시스템 호출(fork)을 이용하여 새로운 프로세스를 생성시키고, 수행중이던 프로세스를 대기(wait)시키며, 새로 생성된 프로세스를 이용하여 데이터 입력 윈도우를 가시화시켜 준다. 설계자는 여기에 데이터를 입력 혹은 수정하고 Save나 Cancel 버튼을 클릭하게 되면 필요한 데이터를 대기중이던 모프로세스에 넘겨주며, 프로그램의 제어가 새로 생성된 프로세스에서 벗어나 모프로세스로 돌아오게 되고, 그 다음 루틴을 계속 수행하게 된다.

6. 결 론

이 연구의 결과를 정리하면 다음과 같다. 첫째로는 전통적인 선박의 개념설계 과정에 처음으로 GUI 개념이 도입되었다. 둘째로는 기존의 선박설계 프로그램에서는 프로그램내에 사용자 인터페이스 부분이 프로그램의 일부로 포함되어 있어 설계 흐름에 제어를 받았지만, 이번에 개발된 MBASWIN은 설계 루틴과 사용자 인터페이스 루틴을 완전 분리하여, 사용자 인터페이스 루틴이 각종 설계루틴들을 제어하도록 함으로써 설계의 유연성을 부여하였다. 셋째로는 특정 GUI Tool을 선정함에 있어서 그래픽 표준이라는 개념이 적용되었다. 넷째로 기존의 절차식 프로그래밍 기법에 의한 설계 프로그램에 Event-driven 방식의 GUI 개념을 도입함으로써 발생하는 문제들을 프로세스간의 통신수단을 통해서 해결할 수 있었다.

이 글에서 소개된 그래픽 사용자 인터페이스는 다음과 같은 좋은 점들을 갖고 있다.

우선, 설계하고자 하는 대상이 가시화되고, 설계 과정 자체도 가시화되므로, 설계자의 생산성을 향상시키게 된다. 또한 대부분의 작업이 마우스로 처리되고 특수한 명령어를 기억할 필요가 없으므로 개발된 시스템을 쉽게 이용할 수 있다. 또한 사용자 인터페이스의 구축에 있어서 표준화된 부품들을 사용하였으므로 다른 EWS 환경에도 쉽게 이식된다. 이것은

장치 독립성 (Device Independent) 또는 이식성 (Portability)이라고 하며, 이 시스템이 CSDP 사업의 일환으로 개발되어 하드웨어 환경이 서로 다른 각 조선소에 Install해야 하는 점을 고려할 때 이와같은 요소들은 매우 중요한 것이 된다.

앞으로 기존의 설계 프로그램을 GUI 개념에 맞게 재편성하고, MBASWIN에서 제공되는 데이터 입력 기능을 좀더 보완한다면, 사용자(설계자) 측면에서 유연성있고 효율적인 선박 개념설계 프로그램이 될 것이다.

참 고 문 헌

- [1] 김연승, 허영환, "News 환경하에서의 과학기술용 전산코드 사용을 위한 User Interface 개발 (prototype 설계 및 구현)", 컴퓨터그래픽스 및 설계자동화 학술발표논문집, 제3권 제1호, 1991. 6.
- [2] Dodani, Mahesh D., Chales E.Hughes, J.Michael Moshell, "Separation of Power", *BYTE*, March, 1989.
- [3] Young, Douglas, A., "*The X Window System : programming and applications with Xi*", Prentice Hall, 1990.
- [4] 한선영, 전길남, "객체지향과 사용자 인터페이스", 정보과학회지, 제8권 제15호, 1990.10.
- [5] 신동우, "조선 CAD/CAM 시스템을 위한 Graphic User Interface 기술", 조선학회 하계강습회 proceeding, 1991.8.
- [6] O'Reilly Tim and Nye, Adrian, "*X Toolkit Intrinsics programming Manual*", Volume 4, O'Reilly & Associates Inc., 1990.
- [7] Sibert, John L., William D.Hurley & Teresa W.Bleser, "An Object-oriented User Interface Management System", *Computer Graphics*, ACM Siggraph, Vol.20, No.4, Aug., 1986.
- [8] 한순홍, 서승완, "컴퓨터 그래픽 표준에 대한 조사 연구", 대한 조선헌회 춘계학술강연회, 1991.4.
- [9] 한순홍, 이동곤, 이경호, "Motif를 이용한 선박 개념설계 시스템용 그래픽 사용자 인터페이스의 개발", 대한기계학회 CAD/CAM 강연회, 1991.
- [10] 김명렬, "C언어 프로그래밍", 산학사, 1986.
- [11] "核心 UNIX 프로그래밍", 홍릉과학출판사 편집부역, 1989.